

## RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

### Universal regularization methods

Cartis, Coralie; Gould, Nick I.; Toint, Philippe L.

*Published in:*  
SIAM Journal on Optimization

*DOI:*  
[10.1137/16M1106316\\_rfseq1](https://doi.org/10.1137/16M1106316_rfseq1)

*Publication date:*  
2019

*Document Version*  
Publisher's PDF, also known as Version of record

#### [Link to publication](#)

*Citation for published version (HARVARD):*  
Cartis, C, Gould, NI & Toint, PL 2019, 'Universal regularization methods: varying the power, the smoothness and the accuracy', *SIAM Journal on Optimization*, vol. 29, no. 1, pp. 595-615.  
[https://doi.org/10.1137/16M1106316\\_rfseq1](https://doi.org/10.1137/16M1106316_rfseq1)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

## RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

### Universal regularization methods -- varying the power, the smoothness and the accuracy

Cartis, Coralie; Gould, N. I. M.; Toint, Ph

*Published in:*  
SIAM Journal on Optimization

*Publication date:*  
2019

*Document Version*  
Publisher's PDF, also known as Version of record

[Link to publication](#)

*Citation for pulished version (HARVARD):*  
Cartis, C, Gould, NIM & Toint, P 2019, 'Universal regularization methods -- varying the power, the smoothness and the accuracy' SIAM Journal on Optimization, vol. 29, no. 1, pp. 595-615.

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# UNIVERSAL REGULARIZATION METHODS: VARYING THE POWER, THE SMOOTHNESS AND THE ACCURACY\*

CORALIA CARTIS<sup>†</sup>, NICK I. GOULD<sup>‡</sup>, AND PHILIPPE L. TOINT<sup>§</sup>

**Abstract.** Adaptive cubic regularization methods have emerged as a credible alternative to linesearch and trust-region for smooth nonconvex optimization, with optimal complexity amongst second-order methods. Here we consider a general/new class of adaptive regularization methods that use first- or higher-order local Taylor models of the objective regularized by a(ny) power of the step size and applied to convexly constrained optimization problems. We investigate the worst-case evaluation complexity/global rate of convergence of these algorithms, when the level of sufficient smoothness of the objective may be unknown or may even be absent. We find that the methods accurately reflect in their complexity the degree of smoothness of the objective and satisfy increasingly better bounds with improving model accuracy. The bounds vary continuously and robustly with respect to the regularization power and accuracy of the model and the degree of smoothness of the objective.

**Key words.** evaluation complexity, worst-case analysis, regularization methods

**AMS subject classifications.** 90C30, 65K05

**DOI.** 10.1137/16M1106316

**1. Introduction.** We consider the (possibly) convexly constrained optimization problem

$$(1.1) \quad \min_{x \in \mathcal{F}} f(x)$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a smooth, possibly nonconvex, objective and where the feasible set  $\mathcal{F} \subset \mathbb{R}^n$  is closed, convex, and nonempty (for example, the set  $\mathcal{F}$  could be described by simple bounds and both polyhedral and more general convex constraints).<sup>1</sup> Clearly, the case of unconstrained optimization is covered here by letting  $\mathcal{F} = \mathbb{R}^n$ . We are interested in the case when  $f \in \mathcal{C}^{p, \beta_p}(\mathcal{F})$ , namely,  $f$  is  $p$ -times continuously differentiable in  $\mathcal{F}$  with the  $p$ th derivative being Hölder continuous of (unknown) degree  $\beta_p \in [0, 1]$ .<sup>2</sup> We consider adaptive regularization methods applied to problem (1.1) that generate feasible iterates  $x_k$  that are (possibly very) approximate minimizers over  $\mathcal{F}$  of local models of the form

$$m_k(x_k + s) = T_p(x_k, s) + \frac{\sigma_k}{r} \|s\|_2^r,$$

\*Received by the editors December 2, 2016; accepted for publication (in revised form) November 6, 2018; published electronically March 5, 2019.

<http://www.siam.org/journals/siopt/29-1/M110631.html>

<sup>†</sup>Mathematical Institute, University of Oxford, Oxford, OX2 6GG, UK (coralia.cartis@maths.ox.ac.uk).

<sup>‡</sup>Computational Science and Engineering Department, STFC-Rutherford Appleton Laboratory, Chilton OX11 0QX, UK (nick.gould@stfc.ac.uk).

<sup>§</sup>Namur Research Center on Complex Systems (NAXYS), University of Namur, B-5000 Namur, Belgium (philippe.toint@unamur.be).

<sup>1</sup>We are tacitly assuming that the cost of evaluating constraint functions and their derivatives is negligible.

<sup>2</sup>Note that if  $\beta_p > 1$ , then the resulting class of objectives is restricted to multivariate polynomials of degree  $p$ . If  $p = 1$ , we only allow  $\beta_1 \in (0, 1]$ , for reasons to be explained later in the paper.

where  $T_p(x_k, s)$  is the  $p$ th-order Taylor polynomial of  $f$  at  $x_k$  and  $r > p \geq 1$ . The parameter  $\sigma_k > 0$  is adjusted to ensure sufficient decrease in  $f$  occurs when the model value is decreased. In this paper, we derive evaluation complexity bounds for finding first-order critical points of (1.1) using higher-order adaptive regularization methods. Despite the higher order of the models, the model minimization is performed only approximately, generalizing the approach in [3]. The proposed methods also ensure that the steps are “sufficiently long,” in a new way, generalizing ideas in [19]. The ensuing complexity analysis shows the robust interplay of the regularization power  $r$ , the model accuracy  $p$ , and the degree of smoothness  $\beta_p$  of the objective, with some surprising results. In particular, we find that the degree of smoothness of the objective—which is often unknown and is even allowed to be absent here—is accurately reflected in the complexity of the methods, independently of the regularization power, provided the latter is sufficiently large. Furthermore, for all possible powers  $r$ , the methods satisfy increasingly better bounds as the accuracy  $p$  of the models and smoothness level  $\beta_p$  are increased. All bounds vary continuously as a function of the regularization power and smoothness level. Table 4.1 summarizes our complexity bounds.

We now review the existing literature in detail and further clarify our approach, motivation, and contributions. Cubic regularization for the (unconstrained) minimization of  $f(x)$  for  $x \in \mathbb{R}^n$  was proposed independently in [20, 25, 27], with [25] showing it has better global worst-case function evaluation complexity than the method of steepest descent. Extending [25], we proposed some practical variants—adaptive regularization with cubics (ARC) [9]—that satisfy the same complexity bound as the regularization methods in [25], namely at most  $\mathcal{O}(\epsilon^{-\frac{3}{2}})$  evaluations are needed to find a point  $x$  for which

$$(1.2) \quad \|\nabla_x f(x)\| \leq \epsilon,$$

under milder requirements on the algorithm (specifically, inexact model minimization). We further showed in [8, 10] that this complexity bound for ARC is sharp and optimal for a large class of second-order methods when applied to functions with globally Lipschitz-continuous second derivatives. Quadratic regularization, namely, a first-order accurate model of the objective regularized by a quadratic term, has also been extensively studied and shown to satisfy the complexity bound of steepest descent, namely,  $\mathcal{O}(\epsilon^{-2})$  evaluations to obtain (1.2) [22]. It was also shown in [9] that one can loosen the requirement that global Lipschitz continuity of the second derivative holds to just global Hölder continuity of the same derivative with exponent  $\beta_2 \in (0, 1]$ . Then, if one also regularizes the quadratic objective model by the power  $2 + \beta_2$  of the step, involving the (often unknown) Hölder exponent, the resulting method requires  $\mathcal{O}(\epsilon^{-\frac{2+\beta_2}{1+\beta_2}})$  evaluations, which, as a function only of  $\epsilon$ , belongs to the interval  $[\epsilon^{-\frac{3}{2}}, \epsilon^{-2}]$ ; these bounds are sharp and optimal for objectives with corresponding levels of smoothness of the Hessian [10]. Note that this bound also holds if  $\beta_2 = 0$ .

An important related question and extension were answered in [3]: if higher-order derivatives are available, can one improve the complexity of regularization methods? It was shown in [3] that if one considers approximately minimizing an  $(r - 1)$ th-order Taylor model of the objective regularized by the (weighted)  $r$ th power of the (Euclidean) norm of the step in each iteration (so  $r = p + 1$ ), the complexity of the resulting adaptive regularization method is  $\mathcal{O}(\epsilon^{-\frac{r}{r-1}})$  evaluations to obtain (1.2), under

the assumption that the  $(r - 1)$ th derivative tensor is globally Lipschitz continuous. The method proposed in [3] measures the progress of each iteration by comparing the Taylor model decrease (without the regularization term) to that of the true function decrease and only requiring mild approximate (local) minimization of the regularized model. Here, we generalize these higher-order regularization methods from [3] to allow for an arbitrary local Taylor model, an arbitrary regularization power of the step, and varying levels of smoothness of the highest-order derivative in the Taylor model.

The interest in considering relaxations of Lipschitz continuity to Hölder continuity of derivatives comes not only from the needs of some engineering applications (such as flows in gas pipelines [16, section 17] and properties of nonlinear PDE problems [1]), but also in its own right in optimization theory, as a bridging case between the smooth and nonsmooth classes of problems [21, 23]. In particular, a zero Hölder exponent for a Hölder-continuous derivative corresponds to a bounded derivative, and an exponent in  $(0, 1)$  corresponds to a continuous but not necessarily differentiable derivative, while an exponent of 1 corresponds to a Lipschitz continuous derivative that can be differentiated again. For the case of functions with Hölder-continuous gradients, methods have already been devised, and their complexity analyzed, both as a weaker set of assumptions and as an attempt to have a “smooth” transition between the smooth and nonsmooth (convex) problem classes, without knowing a priori the level of smoothness of the gradient (i.e., the Hölder exponent) [15, 23]; even lower complexity bounds are known [21]. In [11] we considered regularization methods applied to nonconvex objectives with Hölder-continuous gradients (with unknown exponent  $\beta_1 \in (0, 1]$ ) that employ a first-order quadratic model of the objective regularized by the  $r$ th power of the step. We showed that the worst-case complexity of the resulting regularization methods varies depending on  $\min\{r, 1 + \beta_1\}$ . In particular, when  $1 < r \leq 1 + \beta_1$ , the methods take at most  $\mathcal{O}(\epsilon^{-\frac{r}{r-1}})$  evaluations/iterations until termination, and otherwise at most  $\mathcal{O}(\epsilon^{-\frac{1+\beta_1}{\beta_1}})$  evaluations/iterations to achieve the same condition. The latter complexity bound reflects the smoothness of the objective’s landscape, without prior knowledge or use of it in the algorithm, and is independent of the regularization power. Here we generalize the approach in [11] to  $p$ th-order Taylor models and find that similar bounds can be obtained. Also, we are able to allow  $\beta_p = 0$  provided  $p \geq 2$ . We note that advances beyond Lipschitz continuity of the derivatives for higher-order regularization methods were also obtained in [12], where a class of problems with discontinuous and possibly infinite derivatives (such as when cusps are present) is analyzed, yielding similar bounds to [3].

Recently, Grapiglia and Nesterov [19] proposed a new cubic regularization scheme that yields a *universal* algorithm in the sense that its complexity reflects the (possibly unknown or even absent) degree of sufficient smoothness of the objective; the approach in [19] addresses the case when  $p = 2$ ,  $r = 3$ , and  $\beta_2 \in [0, 1]$  in our framework. Our ARp algorithm includes a modification in a similar (but not identical) vein to that in [19]. In particular, our approach checks a theoretical condition that carefully monitors the length of the step on each iteration on which the objective is sufficiently decreased. The technique in [19] is different in that it requires a specific/new sufficient decrease condition of the objective on each iteration that makes progress. We generalize the approach in [19] and achieve complexity bounds with similar universal properties for varying  $r$ ,  $p$ , and unknown  $\beta_p \in [0, 1]$ , provided  $r \geq p + \beta_p$ . We are also able to analyze ARp’s complexity in the regime  $p < r \leq p + \beta_p$ , providing continuously varying results with  $r$  and  $\beta_p$ .

Our algorithm can be applied to convexly constrained optimization problems with nonconvex objectives, where the constraint/feasibility evaluations are inexpensive, offering another generalization of proposals in [3, 19], which are presented for the unconstrained case only; we also extend [19] by allowing an inexact subproblem solution.

The structure of the paper is as follows: section 2 describes our main algorithmic framework (ARp), section 3 presents our complexity analysis, while section 4 concludes with a summary of our complexity bounds (see Table 4.1) and a discussion of the results.

**2. A universal adaptive regularization framework: ARp.** Let  $f \in \mathcal{C}^p(\mathcal{F})$ , with  $p$  integer,  $p \geq 1$ ; let  $r \in \mathbb{R}$ ,  $r > p \geq 1$ . We measure optimality using a suitable continuous first-order criticality measure for (1.1). We define this measure for a general function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  on  $\mathcal{F}$ : for an arbitrary  $x \in \mathcal{F}$ , the criticality measure is given by

$$(2.1) \quad \pi_h(x) \stackrel{\text{def}}{=} \|P_{\mathcal{F}}[x - \nabla_x h(x)] - x\|,$$

where  $P_{\mathcal{F}}$  denotes the orthogonal projection onto  $\mathcal{F}$ , and  $\|\cdot\|$  the Euclidean norm. Letting  $h(x) := f(x)$  in (2.1), it is known that  $x$  is a first-order critical point of problem (1.1) if and only if  $\pi_f(x) = 0$ . Also note that

$$\pi_f(x) = \|\nabla_x f(x)\| \quad \text{whenever } \mathcal{F} = \mathbb{R}^n.$$

For more properties of this measure see [2, 13].

Our ARp algorithm generates feasible iterates  $x_k$  that (possibly very) approximately minimize the local model

$$(2.2) \quad m_k(x_k + s) = T_p(x_k, s) + \frac{\sigma_k}{r} \|s\|^r \quad \text{subject to } x_k + s \in \mathcal{F},$$

which is a regularization of the  $p$ th-order Taylor model of  $f$  around  $x_k$ ,

$$(2.3) \quad T_p(x_k, s) = f(x_k) + \sum_{j=1}^p \frac{1}{j!} \nabla_x^j f(x_k)[s]^j,$$

where  $\nabla_x^j f(x_k)[s]^j$  is the  $j$ th-order tensor  $\nabla_x^j f(x_k)$  of  $f$  at  $x_k$  applied to the vector  $s$  repeated  $j$  times. Note that  $T_p(x_k, 0) = f(x_k)$ . We will also use the measure (2.1) with  $h(s) := m_k(x_k + s)$  for terminating the approximate minimization of  $m_k(x_k + s)$ , and for which we have again

$$\pi_{m_k}(x_k + s) = \|\nabla_s m_k(x_k + s)\| \quad \text{whenever } \mathcal{F} = \mathbb{R}^n.$$

A summary of the main algorithmic framework is as follows.

**Algorithm 2.1. A universal ARp variant.**

**Step 0: Initialization.** An initial point  $x_0 \in \mathcal{F}$  and an initial regularization parameter  $\sigma_0 \geq 0$  are given, as well as an accuracy level  $\epsilon > 0$ . The constants  $\eta_1, \eta_2, \gamma_1, \gamma_2, \gamma_3, \theta, \sigma_{\min}$ , and  $\alpha$  are also given and satisfy

$$(2.4) \quad \theta > 0, \quad \sigma_{\min} \in (0, \sigma_0], \quad 0 < \eta_1 \leq \eta_2 < 1, \quad \text{and}$$

$$(2.5) \quad 0 < \gamma_3 < 1 < \gamma_1 < \gamma_2 \quad \text{and} \quad \alpha \in (0, \tfrac{1}{3}].$$

Compute  $f(x_0)$ ,  $\nabla_x f(x_0)$  and set  $k = 0$ . If  $\pi_f(x_0) < \epsilon$ , terminate. Else, for  $k \geq 0$ , do:

**Step 1: Model set-up.** Compute derivatives of  $f$  of order 2 to  $p$  at  $x_k$ .

**Step 2: Step calculation.** Compute the step  $s_k$  by approximately minimizing the model  $m_k(x_k + s)$  in (2.2) over  $x_k + s \in \mathcal{F}$  such that the following conditions hold:

$$(2.6) \quad x_k + s_k \in \mathcal{F},$$

$$(2.7) \quad m_k(x_k + s_k) < f(x_k),$$

and

$$(2.8) \quad \pi_{m_k}(x_k + s_k) \leq \theta \|s_k\|^{r-1}.$$

**Step 3: Test for termination.** Compute  $\nabla_x f(x_k + s_k)$ . If  $\pi_f(x_k + s_k) < \epsilon$ , terminate with the approximate solution  $x_\epsilon = x_k + s_k$ .

**Step 4: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define

$$(2.9) \quad \rho_k = \frac{f(x_k) - f(x_k + s_k)}{f(x_k) - T_p(x_k, s_k)}.$$

If  $\rho_k \geq \eta_1$ , check whether

$$(2.10) \quad \sigma_k \|s_k\|^{r-1} \geq \alpha \pi_f(x_k + s_k).$$

If both  $\rho_k \geq \eta_1$  and (2.10) hold, then define  $x_{k+1} = x_k + s_k$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 5: Regularization parameter update.** Set

$$(2.11) \quad \sigma_{k+1} \in \begin{cases} [\max(\sigma_{\min}, \gamma_3 \sigma_k), \sigma_k] & \text{if } \rho_k \geq \eta_2 \text{ and (2.10) holds,} \\ [\sigma_k, \gamma_1 \sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2) \text{ and (2.10) holds,} \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] & \text{if } \rho_k < \eta_1 \text{ or (2.10) fails.} \end{cases}$$

Increment  $k$  by one, and go to Step 1 if  $\rho_k \geq \eta_1$  and (2.10) holds, and to Step 2 otherwise.

Iterations for which  $\rho_k \geq \eta_1$  and (2.10) holds (and so  $x_{k+1} = x_k + s_k$ ) are called *successful* and those for which  $\rho_k \geq \eta_2$  and (2.10) holds are referred to as *very successful*, while the remaining ones are *unsuccessful*. For a(ny)  $j \geq 0$ , we denote the

set of successful iterations up to  $j$  by  $\mathcal{S}_j = \{0 \leq k \leq j : \rho_k \geq \eta_1 \text{ and (2.10) holds}\}$  and the set of unsuccessful ones by  $\mathcal{U}_j = \{0, \dots, j\} \setminus \mathcal{S}_j$ . We have the following simple lemma that relates the number of successful and unsuccessful iterations, and that is ensured by the mechanism of Algorithm 2.1.

LEMMA 2.1 (see [9, Theorem 2.1]). *For any fixed  $j \geq 0$  until termination, let  $\sigma_{\text{up}} > 0$  be such that  $\sigma_k \leq \sigma_{\text{up}}$  for all  $k \leq j$  in Algorithm 2.1. Then*

$$(2.12) \quad |\mathcal{U}_j| \leq \frac{|\log \gamma_3|}{\log \gamma_1} |\mathcal{S}_j| + \frac{1}{\log \gamma_1} \log \left( \frac{\sigma_{\text{up}}}{\sigma_0} \right),$$

where  $|\cdot|$  denotes the cardinality of the respective index set.

*Proof.* The proof of (2.12) follows identically to that in the given reference; note that the sets  $\mathcal{S}_j$  and  $\mathcal{U}_j$  are not identical to the usual ARC ones in [9], but the mechanism for modifying  $\sigma_k$  in ARp coincides with the one in ARC on these iterations, which why the proof of this lemma follows identically to [9, Theorem 2.1].  $\square$

Now we comment on the construction of the ARp algorithm. Note that the model minimization conditions (Step 2) and the definition of  $\rho$  in Step 4 are straightforward generalizations of the approach in [3] to  $p$ th-order Taylor models regularized by different powers  $r$  of the norm of the step. Furthermore, recall that conditions (2.6), (2.7) and (2.8) are approximate *local* optimality conditions for the nonconvex polynomial model  $m_k(x_k + s)$  minimization over a convex set  $x_k + s \in \mathcal{F}$ ; in fact, they are even weaker than that as they require a strict decrease (from the base point  $s = 0$ ) and *approximate first-order criticality* for the convexly constrained model. Thus, any descent optimization method—even first-order algorithms such as the projected gradient method—can be applied to ensure these conditions with ease (and no additional derivatives evaluations are required other than those needed to set up the model  $m_k$  at  $x_k$ ). Designing efficient techniques specifically for the approximate minimization of such regularized, nonconvex, high-order polynomial optimization problems is beyond our scope here but an essential component of the success of such methods. Existing regularization-related approaches are available for general nonconvex problems up to third order [4, 5], or dedicated to convex regularized tensor models (see [24] and the references therein) or specialized to nonlinear least-squares problems [17, 18]; these complement classical references such as [26], where third- and fourth-order tensor methods were proposed.

However, there are two main differences to the now standard approaches to (cubic or higher order) regularization methods. Firstly, we check whether the gradient goes below  $\epsilon$  at each trial point, and, if so, terminate on possibly unsuccessful iterations (Step 3). Secondly, when the step  $s_k$  provides a sufficient decrease according to (2.9), we check whether  $s_k$  satisfies (2.10), and only allow steps that have such a carefully monitored length to be taken by the algorithm; if (2.10) fails or  $\rho_k \leq \eta_1$ ,  $\sigma_k$  is increased. Note that, though the length of the step  $s_k$  decreases as  $\sigma_k$  is increased, this is not the case for the expression  $\sigma_k \|s_k\|^{r-1}$  in (2.10), which increases with  $\sigma_k$ , as Lemma 3.4 implies. These two additional ingredients—the gradient calculation at each trial point and the step length condition (2.10)—are directly related to trying to achieve universality of ARp, extending ideas from [19]. Further explanations and discussions for the theoretical need, or otherwise, for condition (2.10) are given next, in Remark 2.1, and later in the paper, in Remarks 3.2(b) and 3.4(b).

Remark 2.1. We further comment on condition (2.10), its connections to [19] and existing literature, and possible alternatives.



(a) We can replace condition (2.10) with the weaker requirement that  $\sigma_k \|s_k\|^{r-1} \geq \alpha\epsilon$ ; then, all subsequent results would remain unchanged. This choice, however, would make the algorithm construction dependent on the accuracy  $\epsilon$  (in places other than in the termination condition), which is not numerically advisable.

(b) Instead of requiring (2.10) on each successful step, we could ask that each model minimization step calculated in Step 2 satisfies (2.10); if (2.10) failed,  $\sigma_k$  would be increased at the end of Step 2 and the model minimization step would be repeated. This approach may result in an unnecessarily small step in practice, but the ensuing ARp complexity bounds would remain qualitatively similar.

(c) Condition (2.10) does not appear as such in the algorithmic variants proposed in [19], as those enforce sufficient decrease conditions on  $f$  in the algorithm for the case when  $p = 2$  and  $r = 3$ , which is the only case addressed in [19]. But (2.10) (with  $r = 3$ ) is a necessary ingredient for achieving the required sufficient decrease conditions in [19]; see [19, Lemma 2.3] (in particular, equation (2.21)).

(d) Following [19], instead of (2.10), we could employ a different definition of  $\rho_k$  in (2.9), namely, replacing the denominator in (2.9) by a rational function in  $\epsilon$  and  $\sigma_k$ , or by a function of  $\sigma_k$  and the gradient at the new point (see, for example, [19, equation (6.5)]), to achieve the desired order of model/function decrease for universal complexity and behavior. According to our calculations, qualitatively similar complexity bounds would again be obtained for such ARp variants.

We note that specific  $\rho_k$  definitions (namely, those with a denominator connected to the length of the step) that enforce a particular sufficient decrease property for the objective evaluations were also used in [14, 6] for trust-region and quadratic regularization variants, in order to achieve optimal complexity bounds for the ensuing methods.

(e) According to our calculations, without the condition (2.10) on the length of the step, or a similar measure of progress, the complexity of ARp would dramatically (but continuously) worsen in the regime when  $r > p + \beta_p$ , as  $r$  increases. But, as we clarify at the end of section 3, for the case when  $r \leq p + \beta_p$ , same-order complexity bounds could be obtained for ARp without using (2.10); so in principle, for this parameter regime, (2.10) could be removed from the construction of ARp. However, note that, as  $\beta_p$  is not generally known a priori, the regime of most interest—in terms of both best complexity bounds and practicality—is when  $r$  is large, hence the need for condition (2.10) in ARp, for both regimes.

### 3. Worst-case complexity analysis of ARp.

**3.1. Some preliminary properties.** We have the following simple consequence of (2.7).

LEMMA 3.1. *On each iteration of Algorithm 2.1, we have the decrease*

$$(3.1) \quad f(x_k) - T_p(x_k, s_k) \geq \frac{\sigma_k}{r} \|s_k\|^r.$$

*Proof.* Note that condition (2.7) and the definition of  $m_k(s)$  in (2.2) immediately give (3.1).  $\square$

We have the following upper bound on  $s_k$ .

LEMMA 3.2. *On each iteration of Algorithm 2.1, we have*

$$(3.2) \quad \|s_k\| \leq \max_{1 \leq j \leq p} \left\{ \left( \frac{pr}{j! \sigma_k} \|\nabla_x^j f(x_k)\| \right)^{\frac{1}{r-j}} \right\}.$$

*Proof.* It follows from (2.7), (2.2), and (2.3) that

$$s_k^T \nabla_x f(x_k) + \frac{1}{2} \nabla_x^2 f(x_k)[s_k, s_k] + \cdots + \frac{1}{p!} \nabla_x^p f(x^k)[s_k, s_k, \dots, s_k] + \frac{\sigma_k}{r} \|s_k\|^r < 0,$$

which, from Cauchy–Schwarz and norm properties, further implies

$$-\|s_k\| \cdot \|\nabla_x f(x_k)\| - \frac{1}{2} \|s_k\|^2 \cdot \|\nabla_x^2 f(x_k)\| - \cdots - \frac{1}{p!} \|s_k\|^p \cdot \|\nabla_x^p f(x^k)\| + \frac{\sigma_k}{r} \|s_k\|^r < 0$$

or, equivalently,

$$\sum_{j=1}^p \left( \frac{\sigma_k}{p^r} \|s_k\|^r - \frac{1}{j!} \|s_k\|^j \cdot \|\nabla_x^j f(x^k)\| \right) < 0.$$

The inequality above cannot hold unless at least one of the terms on the left-hand side is negative, which is equivalent to (3.2), using also that  $r > p \geq 1$ .  $\square$

Let us assume that  $f \in \mathcal{C}^{p, \beta_p}$ , namely,

(A.1)  $f \in C^p(\mathcal{F})$  and  $\nabla_x^p f$  is Hölder continuous on the path of the iterates and trial points; i.e.,

$$\|\nabla_x^p f(y) - \nabla_x^p f(x_k)\|_T \leq (p-1)! L_p \|y - x_k\|^{\beta_p}$$

holds for all  $y \in [x_k, x_k + s_k]$ ,  $k \geq 0$ , and some constants  $L_p \geq 0$  and  $\beta_p \in [0, 1]$ , where  $\|\cdot\|$  is the Euclidean norm on  $\mathbb{R}^n$  and  $\|\cdot\|_T$  is recursively induced by this norm on the space of the  $p$ th-order tensors.

A simple consequence of (A.1) is that

$$(3.3) \quad |f(x_k + s_k) - T_p(x_k, s_k)| \leq \frac{L_p}{p} \|s_k\|^{p+\beta_p}, \quad k \geq 0,$$

and

$$(3.4) \quad \|\nabla_x f(x_k + s_k) - \nabla_s T_p(x_k, s_k)\| \leq L_p \|s_k\|^{p+\beta_p-1}, \quad k \geq 0;$$

see [3] for a proof of (3.3) and (3.4), with (A.1) replacing Lipschitz continuity of the  $p$ th derivative.

*Remark 3.1.* Note that throughout the paper we assume  $r > p \geq 1$ ,  $r \in \mathbb{R}$ , and  $p \in \mathbb{N}$ ; and that either  $p \geq 1$  and  $\beta_p \in (0, 1]$  or  $p \geq 2$  and  $\beta_p \in [0, 1]$ . Thus, in both cases,  $p + \beta_p - 1 > 0$ .

Two useful preliminary lemmas follow.

LEMMA 3.3. Assume that (A.1) holds. Then, on each iteration of Algorithm 2.1, we have

$$(3.5) \quad \pi_f(x_k + s_k) \leq L_p \|s_k\|^{p+\beta_p-1} + (\sigma_k + \theta) \|s_k\|^{r-1}.$$

*Proof.* Using the triangle inequality and (2.1) with  $h \stackrel{\text{def}}{=} f$  and  $h \stackrel{\text{def}}{=} m_k$ , we obtain

$$\begin{aligned} \pi_f(x_k + s_k) &= \|P_{\mathcal{F}}[x_k + s_k - \nabla_x f(x_k + s_k)] - P_{\mathcal{F}}[x_k + s_k - \nabla_s m_k(x_k + s_k)] \\ &\quad + P_{\mathcal{F}}[x_k + s_k - \nabla_s m_k(x_k + s_k)] - (x_k + s_k)\| \\ &\leq \|P_{\mathcal{F}}[x_k + s_k - \nabla_x f(x_k + s_k)] - P_{\mathcal{F}}[x_k + s_k - \nabla_s m_k(x_k + s_k)]\| \\ &\quad + \pi_{m_k}(x_k + s_k). \end{aligned}$$

The last inequality, the contractive property of the projection operator  $P_{\mathcal{F}}$ , and the inner termination condition (2.8) give

$$(3.6) \quad \pi_f(x_k + s_k) \leq \|\nabla_x f(x_k + s_k) - \nabla_s m_k(x_k + s_k)\| + \theta \|s_k\|^{r-1}.$$

We have from (2.2) that

$$\nabla_s m_k(x_k + s) = \nabla_s T_p(x_k, s) + \sigma_k \|s\|^{r-1} \frac{s}{\|s\|}$$

and so

$$(3.7) \quad \begin{aligned} \|\nabla_x f(x_k + s_k) - \nabla_s m_k(x_k + s_k)\| &\leq \|\nabla_x f(x_k + s_k) - \nabla_s T_p(x_k, s_k)\| + \sigma_k \|s_k\|^{r-1} \\ &\leq L_p \|s_k\|^{p+\beta_p-1} + \sigma_k \|s_k\|^{r-1}, \end{aligned}$$

where we used (3.4) to obtain the second inequality. Now (3.5) follows from replacing (3.7) in (3.6).  $\square$

LEMMA 3.4. Assume that (A.1) holds. If

$$(3.8) \quad \sigma_k \geq \max\{\theta, \kappa_2 \|s_k\|^{p+\beta_p-r}\},$$

where

$$(3.9) \quad \kappa_2 \stackrel{\text{def}}{=} \frac{rL_p}{p(1-\eta_2)},$$

then both  $\rho_k \geq \eta_2$  and (2.10) hold, and so iteration  $k$  is very successful.

*Proof.* We assume that (3.8) holds, which implies that

$$(3.10) \quad \sigma_k \geq \kappa_2 \|s_k\|^{p+\beta_p-r}.$$

The definition of  $\rho_k$  in (2.9) gives

$$|\rho_k - 1| = \frac{|f(x_k + s_k) - T_p(x_k, s_k)|}{f(x_k) - T_p(x_k, s_k)},$$

whose numerator we upper bound by (3.3), and whose denominator we lower bound by (3.1), to deduce

$$(3.11) \quad |\rho_k - 1| \leq \frac{\frac{L_p}{p} \|s_k\|^{p+\beta_p}}{\frac{\sigma_k}{r} \|s_k\|^r} = \frac{rL_p}{p\sigma_k} \|s_k\|^{p+\beta_p-r}.$$

In (3.11), we employ (3.10) and the expression of  $\kappa_2$  in (3.9) to deduce that  $|1 - \rho_k| \leq 1 - \eta_2$ , which ensures that  $\rho_k \geq \eta_2$ .

It remains to show that (3.8) also implies (2.10). From (3.8), we have that  $\sigma_k \geq \theta$ , which, together with (3.5), gives

$$(3.12) \quad \pi_f(x_k + s_k) \leq \|s_k\|^{p+\beta_p-1} (L_p + 2\sigma_k \|s_k\|^{r-p-\beta_p}).$$

The definition (3.9) and requirements  $r > p$  and  $\eta_2 \in (0, 1)$  imply that  $L_p \leq \kappa_2$ . This and (3.12) give

$$(3.13) \quad \pi_f(x_k + s_k) \leq \|s_k\|^{p+\beta_p-1} (\kappa_2 + 2\sigma_k \|s_k\|^{r-p-\beta_p}).$$

From (3.10),  $\kappa_2 \leq \sigma_k \|s_k\|^{r-p-\beta_p}$ . We use this to bound  $\kappa_2$  in (3.13), which gives the inequality

$$\pi_f(x_k + s_k) \leq \|s_k\|^{p+\beta_p-1} (3\sigma_k \|s_k\|^{r-p-\beta_p}) = 3\sigma_k \|s_k\|^{r-1}.$$

Thus,  $\sigma_k \|s_k\|^{r-1} \geq \frac{1}{3} \pi_f(x_k + s_k)$ , which implies (2.10) since  $\alpha \leq \frac{1}{3}$ .  $\square$

**3.2. The case when  $r > p + \beta_p$ .** Using Lemmas 3.3 and 3.4, we have the following result, which, together with its proof, was inspired by and generalizes the result and proof in [19, Lemma 2.3].

LEMMA 3.5. *Let  $r > p + \beta_p$  and assume (A.1). While Algorithm 2.1 has not terminated, if*

$$(3.14) \quad \sigma_k \geq \max \left\{ \theta, \kappa_1 \epsilon^{\frac{p+\beta_p-r}{p+\beta_p-1}} \right\},$$

where

$$(3.15) \quad \kappa_1 \stackrel{\text{def}}{=} (3^{r-p-\beta_p} \kappa_2^{r-1})^{\frac{1}{p+\beta_p-1}} \quad \text{and} \quad \kappa_2 \text{ is defined in (3.9),}$$

then (3.8) holds, and so iteration  $k$  is very successful.

*Proof.* We will prove our result by contradiction. We assume that (3.8) does not hold on iteration  $k$ , and so

$$(3.16) \quad \sigma_k \|s_k\|^{r-p-\beta_p} < \kappa_2.$$

Note that while Algorithm 2.1 does not terminate we have  $\pi_f(x_k + s_k) \geq \epsilon$ . Also, from (3.14),  $\sigma_k \geq \theta$ . We substitute these two inequalities into (3.5) to deduce

$$(3.17) \quad \epsilon \leq L_p \|s_k\|^{p+\beta_p-1} + 2\sigma_k \|s_k\|^{r-1} = \|s_k\|^{p+\beta_p-1} (L_p + 2\sigma_k \|s_k\|^{r-p-\beta_p}).$$

We now employ (3.16) to upper bound the second term in (3.17) by  $2\kappa_2$ , namely,

$$(3.18) \quad \epsilon < \|s_k\|^{p+\beta_p-1} (L_p + 2\kappa_2).$$

We use (3.16) again to provide an upper bound on  $\|s_k\|$ , which is possible since  $r > p + \beta_p$ . Thus,

$$(3.19) \quad \|s_k\| \leq \left( \frac{\kappa_2}{\sigma_k} \right)^{\frac{1}{r-p-\beta_p}}.$$

Using this bound in (3.18), which is possible since  $p + \beta_p > 1$ , we obtain the first inequality below,

$$(3.20) \quad \epsilon < \left( \frac{\kappa_2}{\sigma_k} \right)^{\frac{p+\beta_p-1}{r-p-\beta_p}} (L_p + 2\kappa_2) < \left( \frac{\kappa_2}{\sigma_k} \right)^{\frac{p+\beta_p-1}{r-p-\beta_p}} \cdot (3\kappa_2),$$

where to obtain the second inequality, we use that  $L_p < \kappa_2$ , which in turn follows from (3.9),  $r > p$ , and  $\eta_2 \in (0, 1)$ . Finally, (3.20) and the definition of  $\kappa_1$  in (3.15) imply that  $\sigma_k < \kappa_1 \epsilon^{\frac{p+\beta_p-r}{p+\beta_p-1}}$ , which contradicts (3.14). Thus, (3.8) must hold and Lemma 3.4 implies that  $\rho_k \geq \eta_2$  and (2.10) hold, and so  $k$  is very successful.  $\square$

*Remark 3.2.* (a) (Parameter regime.) The proof of Lemma 3.5 requires  $r > p + \beta_p$  and  $p + \beta_p > 1$  (to deduce (3.19) and (3.20), respectively). However, the result of Lemma 3.5 remains true if  $r = p + \beta_p$ , and it is proved together with the case when  $r < p + \beta_p$  in Lemma 3.10. Note that, when  $r = p + \beta_p$ , (3.14) becomes  $\sigma_k \geq \max\{\theta, \kappa_2\}$ , which precisely matches the corresponding expression (3.32) in Lemma 3.10 for this same case.

(b) (Condition (2.10).) Without employing (2.10), we showed inequality (3.5), which connects the length of the step to that of the projected gradient. The two

terms on the right-hand side of (3.5) have similar forms as powers of  $\|s_k\|$ , with the exponents crucially determined by Hölder continuity properties of the objective and the power of the regularization term in the model, respectively. Lemmas 3.4 and 3.5 proved that if  $\sigma_k$  is sufficiently large, then the second term in (3.5), namely,  $\sigma_k \|s_k\|^{r-1}$ , will be larger than the term that is a multiple of  $\|s_k\|^{p+\beta_p-1}$ , thus ensuring that (2.10) holds. To further explain this point, note that in (3.5), when  $r > p + \beta_p$  and  $\|s_k\| \leq 1$  (which is the difficult case), the larger term on the right-hand side is a multiple of  $\|s_k\|^{p+\beta_p-1}$  when  $\sigma_k$  is larger than a constant. Lemma 3.5 showed that if  $\sigma_k$  is further increased, in an  $\epsilon$ -dependent way, then the term that is a multiple of  $\|s_k\|^{r-1}$  in (3.5) becomes the larger of the two terms.

LEMMA 3.6. *Let  $r > p + \beta_p$  and assume (A.1). Then, while Algorithm 2.1 has not terminated, we have*

$$(3.21) \quad \sigma_k \leq \max \left\{ \sigma_0, \gamma_2 \theta, \gamma_2 \kappa_1 \epsilon^{\frac{p+\beta_p-r}{p+\beta_p-1}} \right\},$$

where  $\kappa_1$  is defined in (3.15).

*Proof.* Let the right-hand side of (3.14) be denoted by  $\bar{\sigma}$ . It follows from Lemma 3.5 and the mechanism of the algorithm that

$$(3.22) \quad \sigma_k \geq \bar{\sigma} \implies \sigma_{k+1} \leq \sigma_k.$$

Thus, when  $\sigma_0 \leq \gamma_2 \bar{\sigma}$ , it follows that  $\sigma_k \leq \gamma_2 \bar{\sigma}$ , where the factor  $\gamma_2$  is introduced for the case when  $\sigma_k$  is less than  $\bar{\sigma}$  and the iteration  $k$  is not very successful. Letting  $k = 0$  in (3.22) gives (3.21) when  $\sigma_0 \geq \gamma_2 \bar{\sigma}$  since  $\gamma_2 > 1$ .  $\square$

We are ready to establish an upper bound on the number of successful iterations until termination.

THEOREM 3.7. *Let  $r > p + \beta_p$ ,  $\epsilon \in (0, 1]$ , and assume (A.1) holds and  $\{f(x_k)\}$  is bounded below by  $f_{\text{low}}$ . Then, for all successful iterations  $k$  until the termination of Algorithm 2.1, we have*

$$(3.23) \quad f(x_k) - f(x_{k+1}) \geq \kappa_{s,p} \epsilon^{\frac{p+\beta_p}{p+\beta_p-1}},$$

where

$$(3.24) \quad \kappa_{s,p} \stackrel{\text{def}}{=} \frac{\eta_1}{r} \left( \frac{\alpha^r}{\sigma_{\max}} \right)^{\frac{1}{r-1}}, \quad \sigma_{\max} \stackrel{\text{def}}{=} \max\{\sigma_0, \gamma_2 \theta, \gamma_2 \kappa_1\},$$

and  $\kappa_1$  is defined in (3.15). Thus, Algorithm 2.1 takes at most

$$(3.25) \quad \left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_{s,p}} \epsilon^{-\frac{p+\beta_p}{p+\beta_p-1}} \right\rceil$$

successful iterations/evaluations of derivatives of degree 2 and above of  $f$  until termination.

*Proof.* On every successful iteration  $k$ , we have  $\rho_k \geq \eta_1$ ; this and Lemma 3.1 imply

$$(3.26) \quad \begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1 (f(x_k) - T_p(x_k, s_k)) \\ &\geq \eta_1 \frac{\sigma_k}{r} \|s_k\|^r = \frac{\eta_1}{r} (\sigma_k \|s_k\|^{r-1}) \|s_k\|. \end{aligned}$$

On every successful iteration  $k$  we also have that (2.10) holds. Thus, while the algorithm has not terminated we have

$$(3.27) \quad \sigma_k \|s_k\|^{r-1} \geq \alpha\epsilon \quad \text{and} \quad \|s_k\| \geq \left(\frac{\alpha\epsilon}{\sigma_k}\right)^{\frac{1}{r-1}}.$$

Substituting the first inequality and then the second inequality in (3.27) into (3.26), we deduce

$$(3.28) \quad f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{r} \alpha\epsilon \|s_k\| \geq \frac{\eta_1}{r} \alpha\epsilon \left(\frac{\alpha\epsilon}{\sigma_k}\right)^{\frac{1}{r-1}} = \frac{\eta_1}{r} \frac{(\alpha\epsilon)^{\frac{r}{r-1}}}{\sigma_k^{\frac{1}{r-1}}}.$$

We use that  $\epsilon \in (0, 1]$  in (3.21) to deduce that

$$(3.29) \quad \sigma_k \leq \sigma_{\max} \epsilon^{\frac{p+\beta_p-r}{p+\beta_p-1}},$$

where  $\sigma_{\max}$  is defined in (3.24). We combine this upper bound with (3.28) to see that

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{r} (\alpha\epsilon)^{\frac{r}{r-1}} \sigma_{\max}^{-\frac{1}{r-1}} \epsilon^{\frac{r-p-\beta_p}{(p+\beta_p-1)(r-1)}} = \frac{\eta_1}{r} \left(\frac{\alpha^r}{\sigma_{\max}}\right)^{\frac{1}{r-1}} \cdot \epsilon^{\frac{p+\beta_p}{p+\beta_p-1}},$$

which gives (3.23). Using that  $f(x_k) = f(x_{k+1})$  on unsuccessful iterations, and that  $f(x_k) \geq f_{\text{low}}$  for all  $k$ , we can sum up over all successful iterations to deduce (3.25).  $\square$

We are left with counting the number of unsuccessful iterations until termination, and the total iteration and evaluation upper bounds.

**LEMMA 3.8.** *Let  $r > p + \beta_p$  and  $\epsilon \in (0, 1]$ . Then, for any fixed  $j \geq 0$  until termination, Algorithm 2.1 satisfies*

$$(3.30) \quad |\mathcal{U}_j| \leq \frac{|\log \gamma_3|}{\log \gamma_1} |\mathcal{S}_j| + \frac{1}{\log \gamma_1} \log \frac{\sigma_{\max}}{\sigma_0} + \frac{r-p-\beta_p}{(p+\beta_p-1)\log \gamma_1} |\log \epsilon|,$$

where  $\sigma_{\max}$  is defined in (3.24).

*Proof.* We apply Lemma 2.1. To prove (3.30), we use  $\epsilon \in (0, 1]$  and the upper bound (3.29) in place of  $\sigma_{\text{up}}$  in (2.12).  $\square$

**COROLLARY 3.9.** *Let  $r > p + \beta_p$ ,  $\epsilon \in (0, 1]$ , and assume that (A.1) holds and that  $\{f(x_k)\}$  is bounded below by  $f_{\text{low}}$ . Then Algorithm 2.1 takes at most*

$$(3.31) \quad \left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_{s,p}} \left(1 + \frac{|\log \gamma_3|}{\log \gamma_1}\right) \epsilon^{-\frac{p+\beta_p}{p+\beta_p-1}} + \frac{r-p-\beta_p}{(p+\beta_p-1)\log \gamma_1} |\log \epsilon| + \frac{1}{\log \gamma_1} \log \frac{\sigma_{\max}}{\sigma_0} \right\rceil$$

iterations/evaluations of  $f$  and its derivatives until termination, where  $\kappa_{s,p}$  and  $\sigma_{\max}$  are defined in (3.24).

*Proof.* The proof follows from Theorem 3.7 and (3.30), where we let  $j$  denote the first iteration with  $\pi_f(x_j + s_j) < \epsilon$  (i.e., the iteration where ARp terminates) and we use  $j = |\mathcal{S}_j| + |\mathcal{U}_j|$ .  $\square$

**Remark 3.3.** (a) (Comment on  $\sigma_{\min}$ .) We note that the lower bound on  $\sigma_k$ ,  $\sigma_k \geq \sigma_{\min} \geq 0$  for all  $k$ , imposed in (2.11), has not been employed in the above proofs and it is also not needed when  $r = p + \beta_p$ . It seems that in the case when  $r \geq p + \beta_p$

such a lower bound on  $\sigma_k$  may follow implicitly from (2.10). However, the requirement involving  $\sigma_{\min} > 0$  is needed for the case when  $r < p + \beta_p$ .

(b) (Comment on  $\epsilon$ .) In our main complexity results (such as Corollary 3.9), we have a restriction on the required accuracy tolerance  $\epsilon \in (0, 1]$ ; this restriction is for simplicity and simplification of expressions, so as to capture dominating terms in the complexity bounds. It is also intuitive, as we think of  $\epsilon$  as (arbitrarily) “small” compared to problem constants. Indeed, instead of an upper bound of 1 on  $\epsilon$ , we could have used a bound depending on problem constants such as  $L_p$ , which would preserve the same dominating terms in the complexity bounds. However, as most such problem constants are generally unknown, we prefer our approach, as it gives the users/readers a concrete value they can use.

The constants in the bound (3.31) and their behavior with respect to increasing values of  $p$  are discussed in section 3.4.

**3.3. The case when  $p < r \leq p + \beta_p$ .** Note that  $p < r \leq p + \beta_p$  imposes that  $\beta_p > 0$  in this case. Also, note that the proof of Lemma 3.5 fails to hold for  $r \leq p + \beta_p$ . Thus, we need a different approach to upper bounding  $\sigma_k$  here. In particular, we need the following additional assumption (for the case when  $r < p + \beta_p$ ).

(A.2) For  $j \in \{1, \dots, p\}$ , the derivative  $\{\nabla^j f(x_k)\}$  is uniformly bounded above with respect to  $k$ , namely,

$$\|\nabla^j f(x_k)\| \leq M_j \text{ for all } k \geq 0, \quad j \in \{1, \dots, p\}.$$

We let

$$M \stackrel{\text{def}}{=} \max_{1 \leq j \leq p} \left\{ \left( \frac{rp}{j! \sigma_{\min}} M_j \right)^{\frac{1}{r-j}} \right\},$$

where  $\sigma_{\min}$  is defined in (2.11).

LEMMA 3.10. *Let  $r \leq p + \beta_p$  and assume (A.1). If  $r < p + \beta_p$ , assume also (A.2) and  $\sigma_{\min} > 0$ . If*

$$(3.32) \quad \sigma_k \geq \max\{\theta, \kappa_2 M^{p+\beta_p-r}\},$$

*where  $\kappa_2$  and  $M$  are defined in (3.9) and (A.2), respectively, then (3.8) holds, and so iteration  $k$  is very successful.*

*Proof.* If  $r = p + \beta_p$ , then (3.32) clearly implies (3.8) and so Lemma 3.4 applies.

If  $r < p + \beta_p$ , then we upper bound  $\|s_k\|$  by using (A.2) in (3.2), as well as  $\sigma_k \geq \sigma_{\min}$ , to deduce that  $\|s_k\| \leq M$ , where  $M$  is defined in (A.2). Now (3.32) implies (3.8) and so Lemma 3.4 again applies, yielding that iteration  $k$  is very successful.  $\square$

We are ready to bound  $\sigma_k$  from above for all iterations.

LEMMA 3.11. *Let  $r \leq p + \beta_p$  and assume (A.1). If  $r < p + \beta_p$ , assume also (A.2) and  $\sigma_{\min} > 0$ . While Algorithm 2.1 has not terminated, we have*

$$(3.33) \quad \sigma_k \leq \max\{\sigma_0, \gamma_2 \theta, \gamma_2 \kappa_2 M^{p+\beta_p-r}\} \stackrel{\text{def}}{=} \sigma_{\text{up}},$$

*where  $\kappa_2$  and  $M$  are defined in (3.9) and (A.2), respectively.*

*Proof.* The proof follows a similar argument to that of Lemma 3.6, with (3.14) replaced by (3.32). Note also that, as  $\epsilon$  does not appear in the bound (3.32), (3.33) yields a constant upper bound on  $\sigma_k$  that is valid for all  $k$ , irrespective of the required accuracy level  $\epsilon$ .  $\square$

We are now ready to upper bound the number of successful iterations of Algorithm 2.1 until termination.

**THEOREM 3.12.** *Let  $r \leq p + \beta_p$ , assume (A.1) and that  $\{f(x_k)\}$  is bounded below by  $f_{\text{low}}$ . If  $r < p + \beta_p$  assume also (A.2) and  $\sigma_{\min} > 0$ . Then for all successful iterations  $k$  until the termination of Algorithm 2.1, we have*

$$(3.34) \quad f(x_k) - f(x_{k+1}) \geq \kappa_{s,r} \epsilon^{\frac{r}{r-1}},$$

where

$$(3.35) \quad \kappa_{s,r} \stackrel{\text{def}}{=} \frac{\eta_1}{r} \left( \frac{\alpha^r}{\sigma_{\text{up}}} \right)^{\frac{1}{r-1}},$$

and  $\sigma_{\text{up}}$  is defined in (3.33). Thus, Algorithm 2.1 takes at most

$$(3.36) \quad \left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_{s,r}} \epsilon^{-\frac{r}{r-1}} \right\rceil$$

successful iterations/evaluations of derivatives of degree 2 and higher of  $f$  until termination.

*Proof.* Note that (3.26), (3.27), and (3.28) continue to hold in this case (they only use general ARp properties and the mechanism of the algorithm). Applying (3.33) in (3.28), we deduce

$$(3.37) \quad f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{r} (\alpha \epsilon)^{\frac{r}{r-1}} \sigma_{\text{up}}^{-\frac{1}{r-1}} = \frac{\eta_1}{r} \left( \frac{\alpha^r}{\sigma_{\text{up}}} \right)^{\frac{1}{r-1}} \cdot \epsilon^{\frac{r}{r-1}},$$

which gives (3.34).

Using that  $f(x_k) = f(x_{k+1})$  on unsuccessful iterations, and that  $f(x_k) \geq f_{\text{low}}$  for all  $k$ , we can sum over all successful iterations to deduce (3.36).  $\square$

We are left with counting the number of total iterations and evaluations.

**COROLLARY 3.13.** *Let  $r \leq p + \beta_p$ , and assume that (A.1) holds and that  $\{f(x_k)\}$  is bounded below by  $f_{\text{low}}$ . If  $r < p + \beta_p$  assume also (A.2) and  $\sigma_{\min} > 0$ . Then Algorithm 2.1 takes at most*

$$(3.38) \quad \left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_{s,r}} \left( 1 + \frac{|\log \gamma_3|}{\log \gamma_1} \right) \epsilon^{-\frac{r}{r-1}} + \frac{1}{\log \gamma_1} \log \frac{\sigma_{\text{up}}}{\sigma_0} \right\rceil$$

iterations/evaluations of  $f$  and its derivatives until termination, where  $\kappa_{s,r}$  and  $\sigma_{\text{up}}$  are defined in (3.36) and (3.33), respectively.

*Proof.* We first upper bound the total number of unsuccessful iterations; for this, we apply Lemma 2.1 to upper bound  $|\mathcal{U}_j|$  with  $\sigma_{\text{up}}$  defined in (3.33). To prove (3.38) holds, use (3.36) and (2.12), where we let  $j$  denote the first iteration with  $\pi_f(x_j + s_j) < \epsilon$  (i.e., the iteration where ARp terminates), and we use  $j = |\mathcal{S}_j| + |\mathcal{U}_j|$ .  $\square$

**Remark 3.4.** (a) (Comment on  $\sigma_{\min}$ .) Note that  $\sigma_{\min} > 0$  only appears/is used in the complexity bounds for the regime  $r < p + \beta_p$  (namely in the definition of the constant  $M$  in (A.2)) and not for the case when  $r = p + \beta_p$  (see also our Remark 3.3 (a)).

(b) (Condition (2.10).) We used (2.10) in the proof of Theorem 3.12 (namely, in the use of (3.28) to deduce (3.37)) and hence for obtaining the main complexity



result in the regime  $p < r \leq p + \beta_p$ . This was, however, not strictly necessary for obtaining same-order complexity bounds (albeit with different constants) in this parameter regime, and was done for simplicity and coherence of the algorithm and results with the regime  $r > p + \beta_p$  (for which (2.10) is needed), and for practicality as  $\beta_p$  is not known a priori. Let us briefly outline how one could bypass the use of (2.10) in the proof of Theorem 3.12. Note first that, in this regime, (2.10) implies, given the constant upper bound (3.33), that  $\|s_k\| \geq \text{constant} \times \epsilon^{\frac{1}{r-1}}$ . A similar lower bound on  $s_k$  can be obtained directly from (3.5) (rather than from (2.10)) as follows: when  $\|s_k\| \leq 1$ , (3.5) implies  $(\sigma_k + \theta + \kappa_2)\|s_k\|^{r-1} \geq \epsilon$ ; thus, using the constant upper bound (3.33) on  $\sigma_k$ ,  $\|s_k\| \geq \min\{1, \text{constant}_{\text{new}} \times \epsilon^{\frac{1}{r-1}}\}$ . Using the latter bound in (3.26), and that  $\sigma_k \geq \sigma_{\min}$  and  $\epsilon \in (0, 1]$ , we can deduce a same-order bound (in  $\epsilon$ ) as in (3.34). This line of proof is remindful of techniques used in [3] (for the case when  $\beta_p = 1$  and  $r = p + 1$ ).

(c) (The Lipschitz continuous case.) Letting  $\beta_p = 1$  (i.e., the  $p$ th-order derivative is Lipschitz continuous) and  $r = p + 1$  recovers the complexity bounds in [3], namely,  $\mathcal{O}(\epsilon^{-\frac{p+1}{p}})$  (albeit with different constants), and shows these bounds continue to hold for any  $r \geq p + 1$ . Note, however, that condition (2.10) is not needed in the ARp algorithm in [3]. Part (b) above explains that (2.10) is not strictly needed for the complexity bounds in the regime  $r \leq p + \beta_p$  (which includes the case when  $\beta_p = 1$  and  $r = p + 1$ ) for our ARp variant, which clarifies the connection with the algorithm in [3].

(d) (The case when  $r = p + \beta_p$ .) Despite their different proofs, when  $r = p + \beta_p$ , the complexity bound (3.38) is *identical* to the (limit of the) bound (3.31). Comparing the expressions of these two bounds, we find that  $r = p + \beta_p$  implies that the  $|\log \epsilon|$  term in (3.31) vanishes, and that the two complexity bounds clearly agree, provided  $\kappa_{s,p} = \kappa_{s,r}$  and  $\sigma_{\max} = \sigma_{\text{up}}$ . Furthermore, the definitions (3.24) and (3.35) trivially imply  $\kappa_{s,p} = \kappa_{s,r}$  if  $\sigma_{\max} = \sigma_{\text{up}}$ . Finally, to see the latter identity, use the corresponding definitions in (3.24) and (3.33) and note that  $r = p + \beta_p$  yields  $\kappa_1 = \kappa_2$ , where  $\kappa_1$  is defined in (3.15).

The constants in the bound (3.38) and their behavior with respect to increasing values of  $p$  are discussed in section 3.4.

**3.4. The constants in the complexity bounds.** In this section we extract the key constants and expressions in the complexity bounds (3.31) and (3.38) with respect to  $p$  and  $r$  and show that in important cases they stay finite as  $p$  grows, for some suitable choices of algorithm parameters.

**The case when  $r = p + 1$ ,  $\beta_p \in [0, 1]$ ,  $p \geq 2$ .** In this case, the complexity bound (3.31) applies for  $\beta_p \in [0, 1]$ . When  $\beta_p = 1$  (the Lipschitz continuous case), the bound (3.38) holds; however, in Remark 3.4(d), we showed that (3.38) and (the limit of) (3.31) coincide when  $r = p + \beta_p = p + 1$ . Hence, without loss of generality, we focus on estimating (3.31) for any  $\beta_p \in [0, 1]$ . Again without prejudice, we ignore algorithm parameters (namely,  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$ ) that are independent of  $p$ , as they can easily be fixed. Then, (3.31) is a constant multiple of

$$(3.39) \quad \left[ \frac{f(x_0) - f_{\text{low}}}{\kappa_{s,p}} \epsilon^{-\frac{p+\beta_p}{p+\beta_p-1}} + \frac{(1-\beta_p)|\log \epsilon|}{p+\beta_p-1} + \log \frac{\sigma_{\max}}{\sigma_0} \right].$$

From (3.9) and (3.15), we deduce

$$(3.40) \quad \kappa_2 = \mathcal{O}(L_p) \quad \text{and} \quad \kappa_1 = 3^{\frac{1-\beta_p}{p+\beta_p-1}} \kappa_2^{\frac{p}{p+\beta_p-1}} = \mathcal{O}\left(L_p^{\frac{p}{p+\beta_p-1}}\right),$$

and hence, from (3.24),

$$(3.41) \quad \sigma_{\max} = \max\{\sigma_0, \gamma_2\theta, \gamma_2\kappa_1\} \quad \text{and} \quad \frac{1}{\kappa_{s,p}} = \mathcal{O}\left((p+1)\sigma_{\max}^{\frac{1}{p}}\right) = \mathcal{O}\left((p+1)\max\left\{\sigma_0^{\frac{1}{p}}, \theta^{\frac{1}{p}}, L_p^{\frac{1}{p+\beta_p-1}}\right\}\right),$$

where we note that the term  $(p+1)$  arises from the denominator of (2.2) and  $r = p+1$ . Note that, for simplicity of calculation, the Hölder constant  $L_p$  in (A.1) was scaled by  $(p-1)!$ . Thus, letting  $L$  denote the usual/unscaled Hölder constant, we have

$$(3.42) \quad L \stackrel{\text{def}}{=} (p-1)!L_p,$$

where we assume that  $L$  is independent or stays bounded with  $p$ . (Of course,  $L$  and  $L_p$  can have further implicit dependencies on  $p$ , which are difficult to make precise.)

Taking (3.42) explicitly into account and using Stirling's formula  $\{(p-1)! \sim [(p-1)/e]^{p-1} \sqrt{2\pi(p-1)}\}$ , we deduce

$$(3.43) \quad \begin{aligned} \lim_{p \rightarrow \infty} (p+1)L_p^{\frac{1}{p+\beta_p-1}} &= \lim_{p \rightarrow \infty} (p+1) \left( \frac{L}{(p-1)!} \right)^{\frac{1}{p+\beta_p-1}} \\ &= \lim_{p \rightarrow \infty} (p+1)L^{\frac{1}{p+\beta_p-1}} [2\pi(p-1)]^{-\frac{1}{2(p+\beta_p-1)}} \left( \frac{p-1}{e} \right)^{-\frac{p-1}{p+\beta_p-1}} \\ &= \lim_{p \rightarrow \infty} \left( \frac{L}{\sqrt{2\pi}} \right)^{\frac{1}{p+\beta_p-1}} \\ &\quad \times \lim_{p \rightarrow \infty} (p+1)(p-1)^{-\frac{1}{2(p+\beta_p-1)}} \left( \frac{p-1}{e} \right)^{-\frac{p-1}{p+\beta_p-1}} \\ &= 1 \times \lim_{p \rightarrow \infty} (p-1)^{-\frac{1}{2(p+\beta_p-1)}} e^{\frac{p-1}{p+\beta_p-1}} \frac{p+1}{(p-1)^{\frac{p-1}{p+\beta_p-1}}} \\ &= 1 \times e \times 1 = e, \end{aligned}$$

where we used the standard limits  $\lim_{u \rightarrow \infty} u^{\frac{1}{u}} = 1$  and  $\lim_{u \rightarrow \infty} c^{\frac{1}{u}} = 1$ , where  $c > 0$  is an arbitrary constant. This and (3.41) imply that

$$\lim_{p \rightarrow \infty} \frac{1}{\kappa_{s,p}} < \infty,$$

provided that

$$(3.44) \quad (p+1)\sigma_0^{\frac{1}{p}} < \infty \quad \text{and} \quad (p+1)\theta^{\frac{1}{p}} < \infty \quad \text{as} \quad p \rightarrow \infty.$$

The limits in (3.44) can be achieved without difficulty by suitable choices/scalings of  $\sigma_0$  and  $\theta$ , which are user-chosen algorithm parameters. In particular, let

$$(3.45) \quad \sigma_0 \stackrel{\text{def}}{=} \frac{\bar{\sigma}_0}{(p-1)!} \quad \text{and} \quad \theta \stackrel{\text{def}}{=} \frac{\bar{\theta}}{(p-1)!}$$

for any constants  $\bar{\sigma}_0$  and  $\bar{\theta}$  independent of  $p$ ; Stirling's formula applied to  $(p-1)!$  and similar calculations to (3.43) can be used to show that (3.45) satisfy (3.44).

The second term in the sum (3.39) either vanishes when  $\beta_p = 1$  or converges to zero as  $p \rightarrow 0$ . Proceeding to the third term in the sum (3.39), we have the following: from (3.40) and (3.42), we deduce  $\kappa_1 \rightarrow 0$  as  $p \rightarrow \infty$  and so, irrespective of the scaling of  $\sigma_0$  and  $\theta$ ,  $1 \leq \sigma_{\max}/\sigma_0 < \infty$ . Thus, the last term in (3.39) is finite.

We can safely conclude now that, as  $p \rightarrow \infty$ , all constants in (3.39) stay bounded or converge to zero for appropriate choices of  $\sigma_0$  and  $\theta$ , and so, using also that  $\epsilon \in (0, 1]$ , the bound (3.31) approaches  $\mathcal{O}(\epsilon^{-1})$ .

The above discussion of limiting constants can be easily extended, with similar results, to any  $r = ap + b$  with  $a, b > 0$  independent of  $p$ , provided  $r > p + \beta_p$ .

Note also that the more practical case is when  $p$  is fixed and  $\epsilon$  can be made arbitrarily small; then, the bound (3.31) is well defined for all algorithm and problem parameter choices, allowing the use of simplified constants and unscaled parameters in the analysis.

**The case when  $r = p + \beta_p$ ,  $\beta_p \in [0, 1]$ ,  $p \geq 2$ .** In this case, the bound (3.38) applies (note that the case when  $\beta_p = 1$  was already addressed in the first case of this section). The constants in (3.38) stay bounded as  $p$  grows, provided  $\sigma_0$  and  $\theta$  are scaled according to (3.45). Indeed, one can show this very similarly to the case when  $r = p + 1$  above, using (3.9), (3.35), and (3.42) to obtain the following estimates:

$$\kappa_2 = \mathcal{O}(L_p) = \mathcal{O}\left(\frac{L}{(p-1)!}\right), \quad \sigma_{\text{up}} = \max\{\sigma_0, \gamma_2\theta, \gamma_2\kappa_2\} = \mathcal{O}(\max\{\sigma_0, \theta, L_p\}).$$

Letting  $r = p + \beta_p$  in (3.35), we have

$$\begin{aligned} \frac{1}{\kappa_{s,r}} &= \mathcal{O}\left(r\sigma_{\text{up}}^{\frac{1}{r-1}}\right) = \mathcal{O}\left((p + \beta_p)\sigma_{\text{up}}^{\frac{1}{p+\beta_p-1}}\right) \\ &= \mathcal{O}\left((p + \beta_p)(\max\{\sigma_0, \theta, L_p\})^{\frac{1}{p+\beta_p-1}}\right) < \infty \quad \text{as } p \rightarrow \infty, \end{aligned}$$

where the limit follows similarly to (3.43), using also (3.45). As  $p$  grows and as a function of  $\epsilon$ , (3.38) approaches the same well-defined limit as (3.31), namely,  $\mathcal{O}(\epsilon^{-1})$ .

**The case when  $p < r < p + \beta_p$ ,  $\beta_p \in [0, 1]$ ,  $p \geq 2$ .** In this case, the bound (3.38) applies. However, the limiting constants in (3.38) depend crucially on  $M$  in (A.2), which grows unbounded with  $p$ .

#### 4. Discussion of complexity bounds.

**4.1. The cubic regularization algorithm.** We now particularize our algorithm and results to the case when  $p = 2$  and  $r = p + 1$ , which yields a cubic regularization model (2.2) and algorithm, with condition (2.10), namely,

$$(4.1) \quad \sigma_k \|s_k\|^2 \geq \alpha \pi_f(x_k + s_k),$$

imposed on any successful step  $s_k$ , and which allows  $\sigma_{\min} = 0$  in (2.11).

**COROLLARY 4.1.** *Let  $p = 2$ ,  $r = 3$ , and  $\epsilon \in (0, 1]$ . Assume that  $f \in C^2(\mathcal{F})$ , and  $\nabla_x^2 f$  is Hölder continuous on the path of the iterates and trial points with exponent  $\beta_2 \in [0, 1]$ . Let  $\{f(x_k)\}$  be bounded below by  $f_{\text{low}}$ . Then, for all successful iterations  $k$  until the termination of Algorithm 2.1, we have*

$$(4.2) \quad f(x_k) - f(x_{k+1}) \geq \kappa_{s,2} \epsilon^{\frac{2+\beta_2}{1+\beta_2}},$$

where

$$(4.3) \quad \kappa_{s,2} \stackrel{\text{def}}{=} \frac{\eta_1}{3} \left( \frac{\alpha^3}{\sigma_{\max}} \right)^{\frac{1}{2}}, \quad \sigma_{\max} \stackrel{\text{def}}{=} \max \{ \sigma_0, \gamma_2 \theta, \gamma_2 \kappa_1 \},$$

and  $\kappa_1 \stackrel{\text{def}}{=} 3^{\frac{3-\beta_2}{1+\beta_2}} \left[ \frac{L_2}{2(1-\eta_2)} \right]^{\frac{2}{1+\beta_2}}$ . Thus, Algorithm 2.1 takes at most

$$(4.4) \quad \left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_{s,2}} \epsilon^{-\frac{2+\beta_2}{1+\beta_2}} \right\rceil$$

successful iterations/evaluations of derivatives of degree 2 of  $f$  until termination, and at most

$$(4.5) \quad \left\lceil \frac{f(x_0) - f_{\text{low}}}{\kappa_{s,2}} \left( 1 + \frac{|\log \gamma_3|}{\log \gamma_1} \right) \epsilon^{-\frac{2+\beta_2}{1+\beta_2}} + \frac{1-\beta_2}{(1+\beta_2)\log \gamma_1} |\log \epsilon| + \frac{1}{\log \gamma_1} \log \frac{\sigma_{\max}}{\sigma_0} \right\rceil$$

iterations/evaluations of  $f$  and its first and second derivatives until termination, where  $\kappa_{s,2}$  and  $\sigma_{\max}$  are defined in (4.3).

*Proof.* Clearly, the results follow from Corollary 3.9 for  $p = 2$ ,  $r = 3$ , and  $\beta_2 \in [0, 1]$ , and from Corollary 3.13 for  $p = 2$ ,  $r = 3$ , and  $\beta_2 = 1$ . We note the key ingredients that are needed to obtain (4.2), with the remaining results following from standard telescopic sum arguments and from Lemma 2.1, respectively. Lemmas 3.6 and 3.11 provide the following upper bound on  $\sigma_k$ :

$$\sigma_k \leq \sigma_{\max} \epsilon^{-\frac{1-\beta_2}{1+\beta_2}}, \quad k \geq 0.$$

On successful steps, this bound and condition (4.1) (which is (2.10)) are then substituted into the objective decrease condition (3.26), which here takes the form

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{3} \sigma_k \|s_k\|^3 \geq \frac{\eta_1}{3} \alpha \epsilon \left( \frac{\alpha \epsilon}{\sigma_k} \right)^{\frac{1}{2}} \geq \frac{\eta_1}{3} \left( \frac{\alpha^3}{\sigma_{\max}} \right)^{\frac{1}{2}} \epsilon^{\frac{3}{2}}. \quad \square$$

The impact of the value of  $\beta_2 \in [0, 1]$  can be seen in the bound (4.5); for example, when  $\beta_2 = 1$ , the  $|\log \epsilon|$  term disappears, in agreement with known bounds for ARC [9]. Note that, as a function of  $\epsilon$ , Corollary 4.1 matches corresponding bounds in [19] (for different cubic regularization variants) and extends them to convex constraints, allowing inexact subproblem solutions. Our purpose here is also to allow  $p \geq 2$ , and a discussion of the bounds we obtained follows.

**4.2. General discussion of the complexity bounds.** Table 4.1 gives a summary of our complexity bounds as a function of  $r$  and  $q$ .

Several remarks and comparisons are in order concerning these bounds.

- *The first-order case.* Note that the case when  $p = 1$  is also covered, with a more general quadratic model and using a Cauchy analysis, in [11]; the same complexity bounds as in Table 4.1 ensue (as a function of the accuracy) for  $p = 1$ ; the case when  $\beta_1 = 0$  is also not covered in [11].

- *Sharpness.* For unconstrained problems ( $\mathcal{F} = \mathbb{R}^n$ ), the bound for the case when  $p = 1$  and  $r \geq 1 + \beta_1$ ,  $\beta_1 \in (0, 1]$ , was shown to be sharp in [11]. Also, the bounds for ARp with  $p = 2$  and  $2 < r \leq 2 + \beta_2$ ,  $\beta_2 \in (0, 1]$ , are sharp and optimal for the corresponding smoothness classes [10]. We also note that, for general  $p$ ,  $r = p + 1$ , and

TABLE 4.1

Summary of complexity bounds for regularization methods for ranges of  $r$ . Recall that we assumed  $\epsilon \in (0, 1]$ ,  $r > p \geq 1$ ,  $r \in \mathbb{R}$ , and  $p \in \mathbb{N}$ ; and either  $p \geq 1$  and  $\beta_p \in (0, 1]$ , or  $p \geq 2$  and  $\beta_p \in [0, 1]$ . Also, the ranges in the second column are functions of the dominating terms in  $\epsilon$  and varying  $r$  in the appropriate interval and plot the changing bound  $\mathcal{O}(\epsilon^{\frac{r}{r-1}})$ .

Algorithm	$p < r \leq p + \beta_p$	$p + \beta_p < r$
ARp with $p = 1$	$\mathcal{O}(\epsilon^{-\frac{r}{r-1}}) = [\mathcal{O}(\epsilon^{-\frac{1+\beta_1}{\beta_1}}), \infty)$	$\mathcal{O}(\epsilon^{-\frac{1+\beta_1}{\beta_1}})$
ARp with $p = 2$	$\mathcal{O}(\epsilon^{-\frac{r}{r-1}}) = [\mathcal{O}(\epsilon^{-\frac{2+\beta_2}{1+\beta_2}}), \mathcal{O}(\epsilon^{-2})]$	$\mathcal{O}(\epsilon^{-\frac{2+\beta_2}{1+\beta_2}})$
ARp with $p = 3$	$\mathcal{O}(\epsilon^{-\frac{r}{r-1}}) = [\mathcal{O}(\epsilon^{-\frac{3+\beta_3}{2+\beta_3}}), \mathcal{O}(\epsilon^{-\frac{3}{2}})]$	$\mathcal{O}(\epsilon^{-\frac{3+\beta_3}{2+\beta_3}})$
$\vdots$	$\vdots$	$\vdots$
ARp with $p \geq 2$	$\mathcal{O}(\epsilon^{-\frac{r}{r-1}}) = [\mathcal{O}(\epsilon^{-\frac{p+\beta_p}{p+\beta_p-1}}), \mathcal{O}(\epsilon^{-\frac{p}{p-1}})]$	$\mathcal{O}(\epsilon^{-\frac{p+\beta_p}{p+\beta_p-1}})$

$\beta_p = 1$  (the Lipschitz continuous case), [7] shows the bounds for (possibly randomized) ARp variants (in [3]) are sharp and optimal. The difficult example functions in [7] increase in dimension with  $p$ , in contrast to the uni- or bivariate examples in [11, 10].

- *Continuity.* All bounds vary continuously with  $r$  and  $\beta_p \in [0, 1]$ . In particular, when  $r = p + \beta_p$ , the complexity bounds in the second and third columns match (for a given  $p$  and  $\beta_p$ ) (see also Remark 3.4(d)).

- *Universality* [21, 23, 19]. For fixed  $p$  and  $\beta_p$ , the best complexity bounds are obtained when  $r \geq p + \beta_p$ . These bounds do not depend on the regularization power  $r$ , and even though the smoothness parameter  $\beta_p$  is (usually) unknown, its value is captured accurately in the complexity, even for the case when  $\beta_p = 0$  and  $p \geq 2$ . Note that the values of the complexity bounds as a function of the accuracy indicate that one should choose  $r \geq p + 1$  to achieve the best complexity when  $\beta_p$  is unknown; and there seems to be little reason, from an evaluation complexity point of view, to pick anything other than  $r = p + 1$ . (But, note that, as a benefit of using (2.10), one can simplify ARp's construction by not imposing a lower bound  $\sigma_{\min}$  in the  $\sigma_k$  update (2.11).)

- *Complexity values in the order of the accuracy.* Table 4.1 shows the increasingly good complexity obtained as  $p$  grows and  $\beta_p \in [0, 1]$ , namely, as more derivatives become available and the smoother these derivatives are. In particular, purely as a function of  $\epsilon$  and as  $r$  varies, we obtain the following ranges of complexity powers:  $[\epsilon^{-2}, \infty)$  ( $p = 1$ );  $[\epsilon^{-\frac{3}{2}}, \epsilon^{-2}]$  ( $p = 2$ );  $[\epsilon^{-\frac{4}{3}}, \epsilon^{-\frac{3}{2}}]$  ( $p = 3$ );  $[\epsilon^{-\frac{5}{4}}, \epsilon^{-\frac{4}{3}}]$  ( $p = 4$ ); and so on.

- *The Lipschitz continuous case.* Letting  $\beta_p = 1$  (namely, the  $p$ th-order derivative is Lipschitz continuous) and  $r = p + 1$  in Table 4.1 recovers the complexity bounds in [3], namely,  $\mathcal{O}(\epsilon^{-\frac{p+1}{p}})$ ; see also Remark 3.4(c). Furthermore, the results here show that for our ARp variant this complexity bound continues to hold for any regularization power  $r \geq p + 1$ .

- *Loss of smoothness.* Note that, for fixed  $p \geq 2$ ,  $\beta_p = 0$  corresponds to the case when the objective has the highest level of nonsmoothness compared to  $\beta_p \in (0, 1]$ . Then ARp can still be applied, and the good complexity bounds for the case when  $r \geq p + \beta_p \geq 2$  hold.

• *Constants in the complexity bounds.* The constants in the complexity bounds for  $r \geq p + \beta_p$  stay bounded (above) as  $p$  grows, provided some user-chosen algorithm parameters are suitably scaled and that  $r = O(p)$  (see section 3.4). Thus, these complexity bounds remain valid with growing  $p$  and approach  $\mathcal{O}(\epsilon^{-1})$ .

**5. Conclusions.** We have generalized and modified the regularization methods in [3] to allow for varying regularization power, accuracy of Taylor polynomials, and different (Hölder) smoothness levels of derivatives. Our results show the robustness of the evaluation complexity bounds with respect to such perturbations. We found that complexity bounds of regularization methods improve with growing accuracy of the Taylor models and increasing smoothness levels of the objective. Furthermore, when the regularization power  $r$  is sufficiently large (say  $r \geq p + 1$ ) our modification to ARp in the spirit of [19] allows ARp's worst-case behavior to be independent of the regularization power and to accurately reflect the (often unknown) smoothness level of the objective. We have also generalized [3, 19] to problems with convex constraints and inexact subproblem solutions. The question as to whether the complexity bounds we obtained are sharp remains open when  $r \neq p + \beta_p$  and  $p \geq 3$ . This question is particularly poignant in the case when  $p < r < p + \beta_p$ : could a suitable modification of ARp achieve an (improved) evaluation complexity bound that is independent of the regularization power in this case as well?

#### REFERENCES

- [1] A. BENSOUSSAN AND J. FREHSE, *Regularity Results for Nonlinear Elliptic Systems and Applications*, Springer, Berlin, 2002.
- [2] D. P. BERTSEKAS, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.
- [3] E. G. BIRGIN, J. L. GARDENGHI, J. M. MARTÍNEZ, S. A. SANTOS, AND PH. L. TOINT, *Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models*, Math. Program., 163 (2017), pp. 359–368.
- [4] E. G. BIRGIN, J. L. GARDENGHI, J. M. MARTÍNEZ, AND S. A. SANTOS, *Remark on Algorithm 566: Modern Fortran Routines for Testing Unconstrained Optimization Software with Derivatives up to Third-Order*, Technical report, Department of Computer Science, University of São Paulo, Brazil, 2018.
- [5] E. G. BIRGIN, J. L. GARDENGHI, J. M. MARTÍNEZ, AND S. A. SANTOS, *On the Use of Third-Order Models with Fourth-Order Regularization for Unconstrained Optimization*, Technical report, Department of Computer Science, University of São Paulo, Brazil, 2018.
- [6] E. G. BIRGIN AND J. M. MARTÍNEZ, *The use of quadratic regularization with a cubic descent condition for unconstrained optimization*, SIAM J. Optim., 27 (2017), pp. 1049–1074.
- [7] Y. CARMON, J. C. DUCHI, O. HINDER, AND A. SIDFORD, *Lower Bounds for Finding Stationary Points*, I, preprint, <https://arxiv.org/abs/1710.11606>, 2017.
- [8] C. CARTIS, N. I. M. GOULD, AND PH. L. TOINT, *On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization*, SIAM J. Optim., 20 (2010), pp. 2833–2852.
- [9] C. CARTIS, N. I. M. GOULD, AND PH. L. TOINT, *Adaptive cubic overestimation methods for unconstrained optimization. Part II: Worst-case function-evaluation complexity*, Math. Program., 130 (2011), pp. 295–319.
- [10] C. CARTIS, N. I. M. GOULD, AND PH. L. TOINT, *Optimal Newton-type Methods for Nonconvex Smooth Optimization Problems*, ERGO technical report 11-009, School of Mathematics, University of Edinburgh, 2011.
- [11] C. CARTIS, N. I. M. GOULD, AND PH. L. TOINT, *Worst-case evaluation complexity of regularization methods for smooth unconstrained optimization using Hölder continuous gradients*, Optim. Methods Softw., 32 (2017), pp. 1273–1298.
- [12] X. CHEN, PH. L. TOINT, AND H. WANG, *Partially Separable Convexly-Constrained Optimization with Non-Lipschitzian Singularities and Its Complexity*, preprint, <https://arxiv.org/abs/1704.06919>, 2017.
- [13] A. R. CONN, N. I. M. GOULD, AND PH. L. TOINT, *Trust Region Methods*, MOS-SIAM Ser. Optim., SIAM, Philadelphia, PA, 2000, doi: 10.1137/1.9780898719857.

- [14] F. E. CURTIS, D. P. ROBINSON, AND M. SAMADI, *A trust region algorithm with a worst-case iteration complexity of  $O(\epsilon^{-3/2})$  for nonconvex optimization*, Math. Program., 162 (2017), pp. 1–32.
- [15] O. DEVOLDER, *Exactness, Inexactness and Stochasticity in First-Order Methods for Large-Scale Convex Optimization*, PhD thesis, ICTEAM and CORE, Université Catholique de Louvain, 2013.
- [16] GAS PROCESSORS AND SUPPLIERS ASSOCIATION, *Engineering Data Book*. Vol. 2, GPSA, Tulsa, OK, 1994.
- [17] N. I. M. GOULD, T. REES, AND J. SCOTT, *A Higher-order Method for Solving Nonlinear Least-squares Problems*, RAL preprint RAL-P-2017-010, STFC Rutherford Appleton Laboratory, Chilton, United Kingdom, 2017.
- [18] N. I. M. GOULD, T. REES, AND J. SCOTT, *Convergence and Evaluation-Complexity Analysis of a Regularized Tensor-Newton Method for Solving Nonlinear Least-squares Problems*, RAL preprint RAL-P-2017-009, STFC Rutherford Appleton Laboratory, Chilton, United Kingdom, 2017.
- [19] G. N. GRAPIGLIA AND YU. NESTEROV, *Regularized Newton Methods for Minimizing Functions with Hölder Continuous Hessians*, SIAM J. Optim., 27 (2017), pp. 478–506.
- [20] A. GRIEWANK, *The Modification of Newton's Method for Unconstrained Optimization by Bounding Cubic Terms*, Technical report NA/12 (1981), Department of Applied Mathematics and Theoretical Physics, University of Cambridge, United Kingdom, 1981.
- [21] A. S. NEMIROVSKI AND D. B. YUDIN, *Problem Complexity and Method Efficiency in Optimization*, Wiley Ser. Discrete Math., Wiley-Interscience, New York, 1983.
- [22] YU. NESTEROV, *Introductory Lectures on Convex Optimization*, Appl. Optim., Kluwer Academic, Dordrecht, The Netherlands, 2004.
- [23] YU. NESTEROV, *Universal gradient methods for convex optimization problems*, Math. Program., 152 (2015), pp. 381–404.
- [24] YU. NESTEROV, *Implementable Tensor Methods in Unconstrained Convex Optimization*, CORE Discussion Paper, Université Catholique de Louvain, Belgium, 2015.
- [25] YU. NESTEROV AND B. T. POLYAK, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.
- [26] R. B. SCHNABEL AND T. T. CHOW, *Tensor methods for unconstrained optimization using second derivatives*, SIAM J. Optim., 1 (1991), pp. 293–315.
- [27] M. WEISER, P. DEUFLHARD, AND B. ERDMANN, *Affine conjugate adaptive Newton methods for nonlinear elastomechanics*, Optim. Methods Softw., 22 (2007), pp. 413–431.